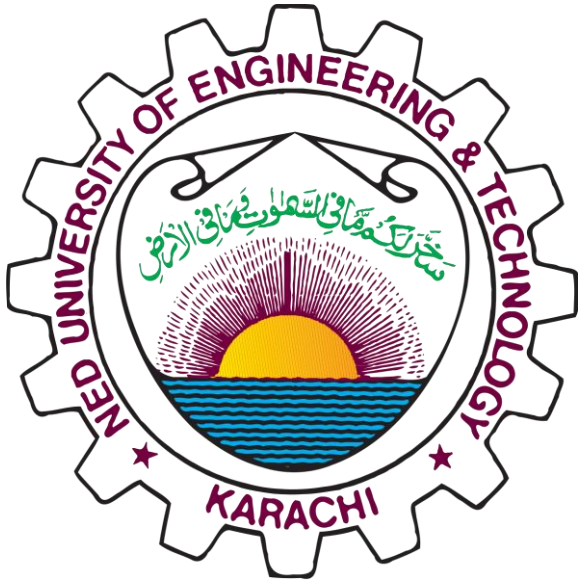
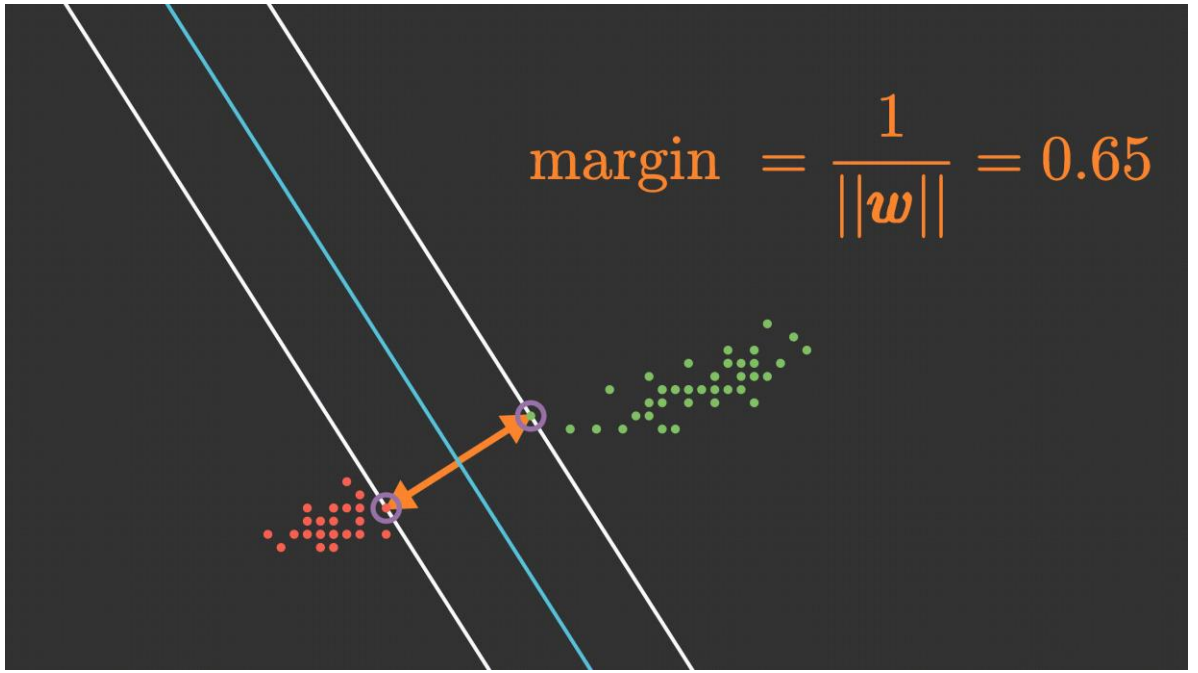


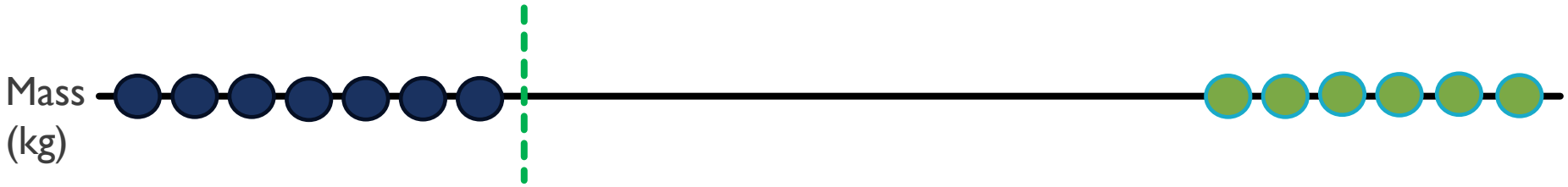
CT-562
MACHINE LEARNING

NED University of Engineering & Technology

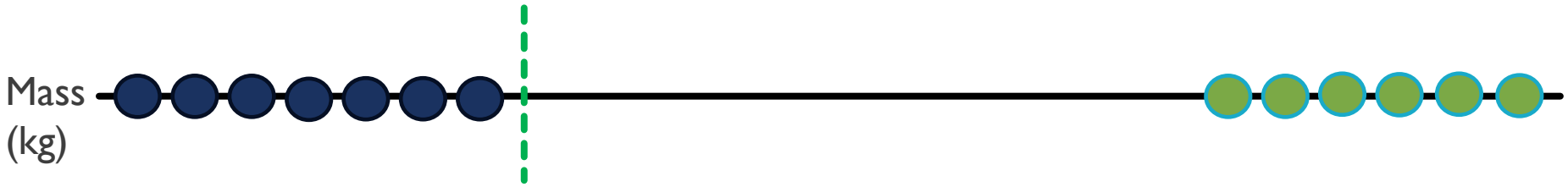


SUPPORT VECTOR MACHINE

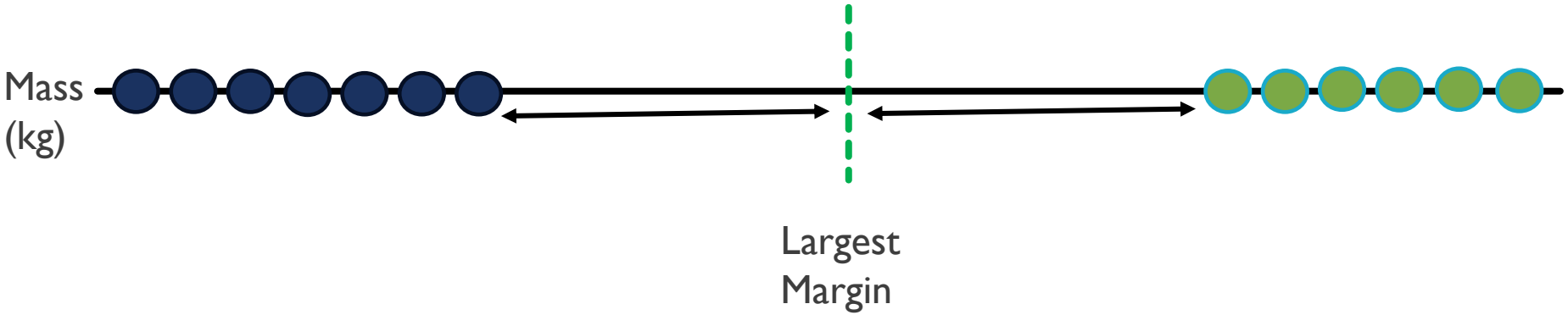
DATA



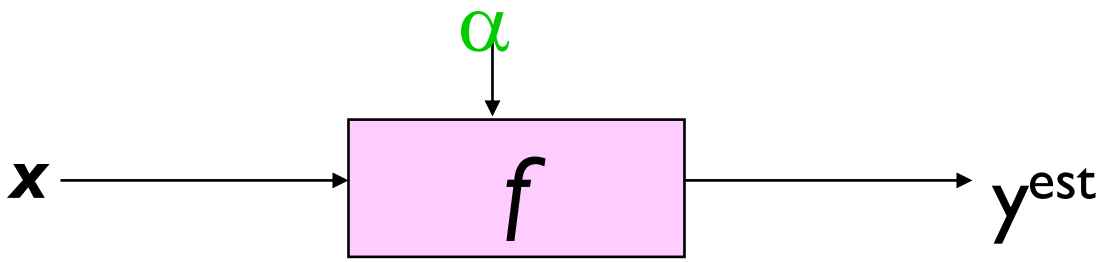
DATA



DATA



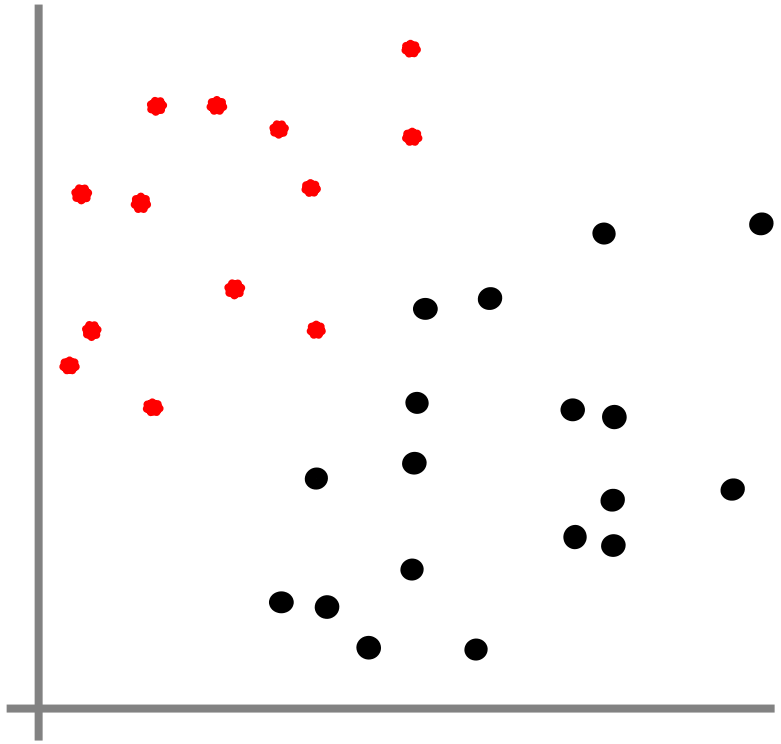
DATA



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} - b)$

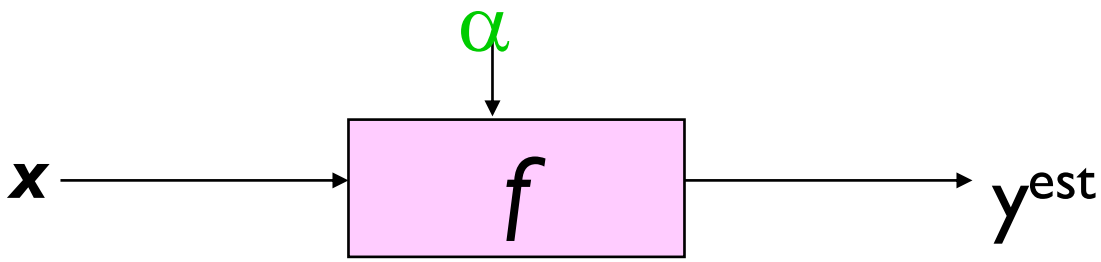
denotes +1

denotes -1



How would you classify this data?

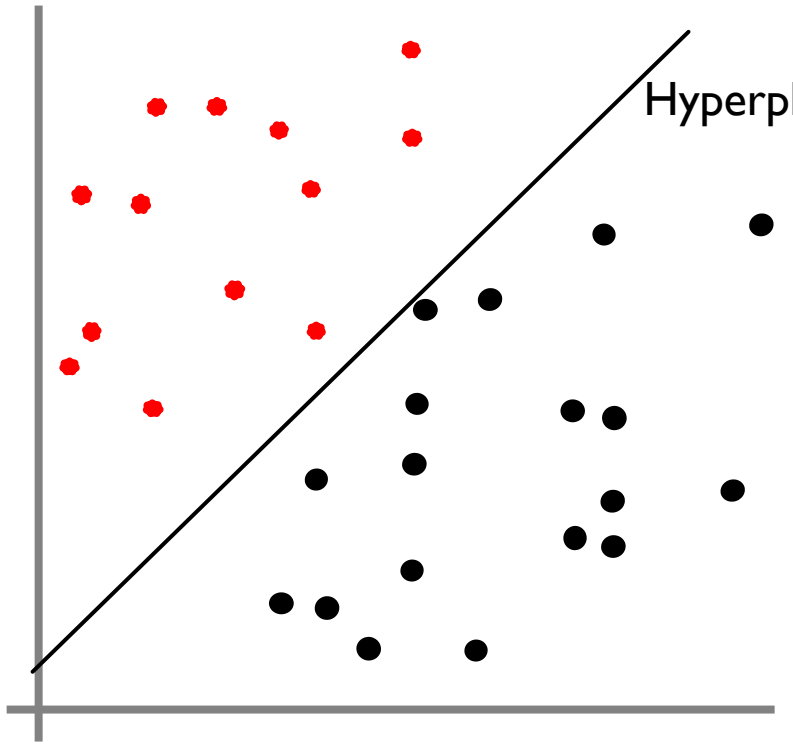
HYPERPLANE



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} - b)$

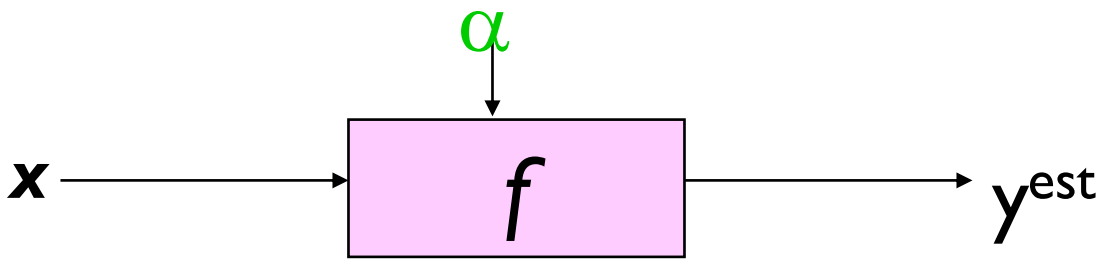
denotes +1

denotes -1



How would you classify this data?

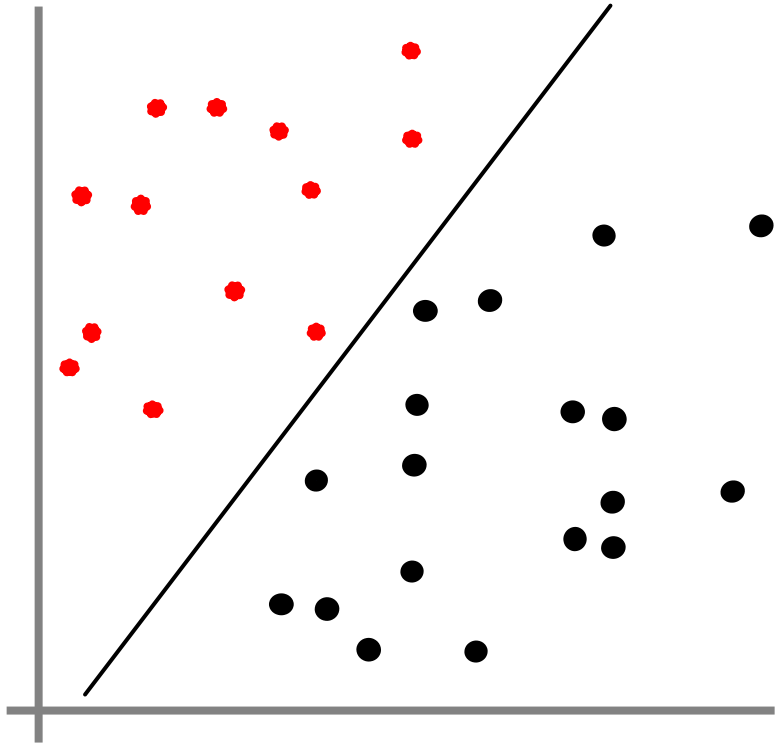
HYPERPLANE



$f(\boldsymbol{x},\boldsymbol{w},b) = sign(\boldsymbol{w}.\boldsymbol{x} - b)$

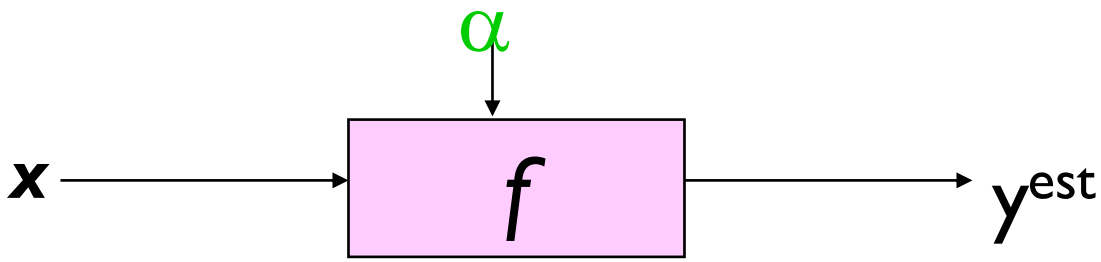
denotes +1

denotes -1



How would you classify this data?

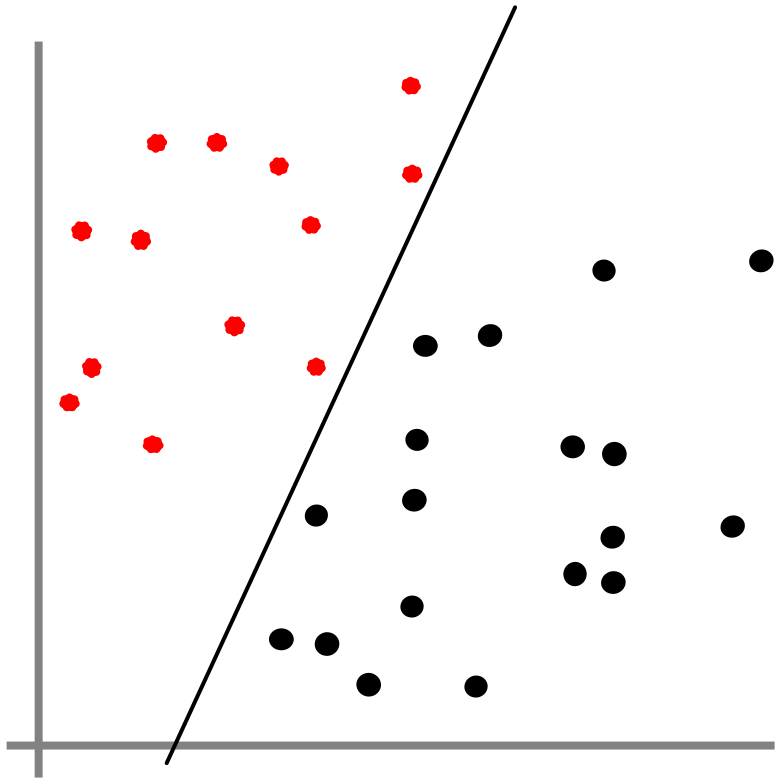
HYPERPLANE



$f(\boldsymbol{x},\boldsymbol{w},b) = sign(\boldsymbol{w}.\boldsymbol{x} - b)$

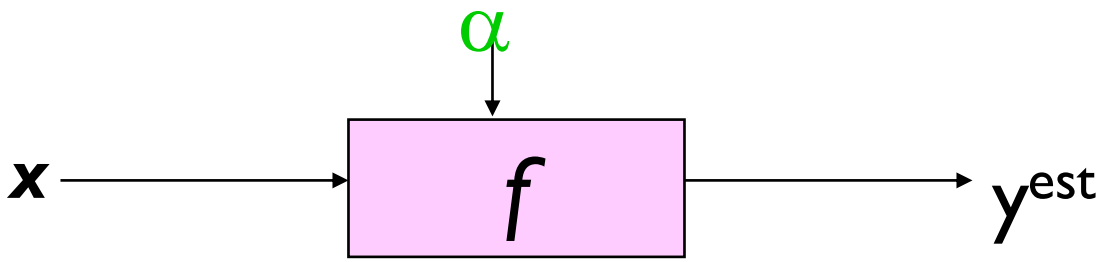
denotes +1

denotes -1



How would you classify this data?

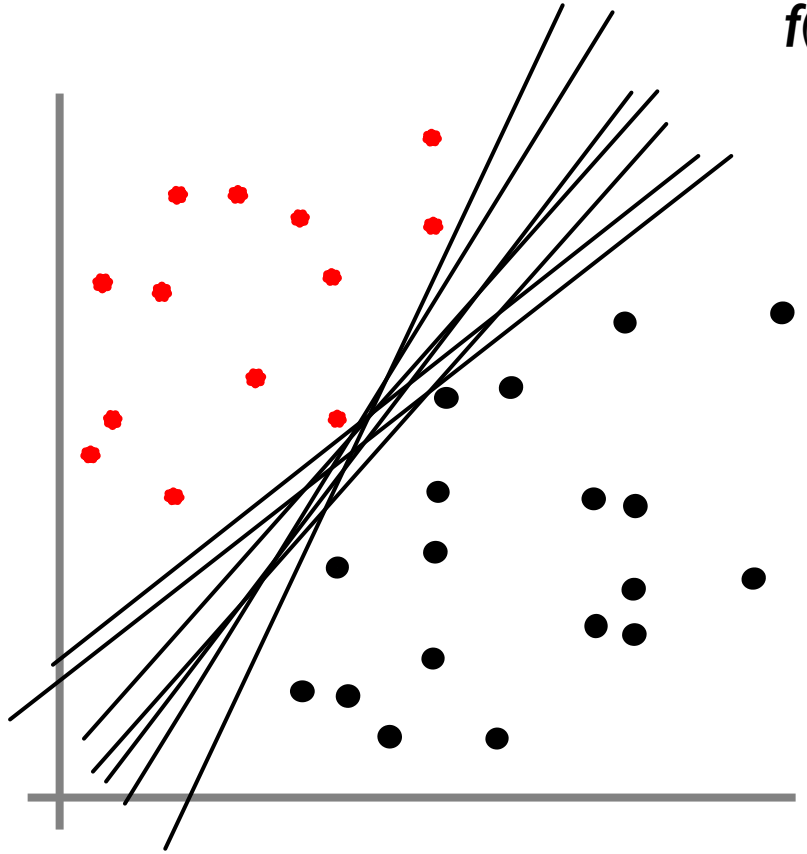
HYPERPLANE



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} - b)$

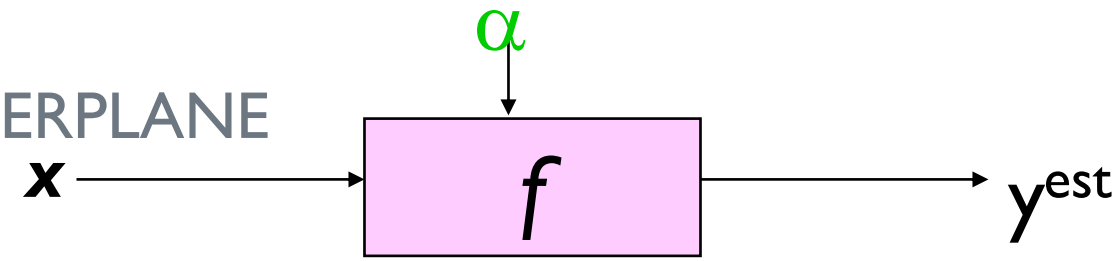
denotes +1

denotes -1



How would you classify this data?

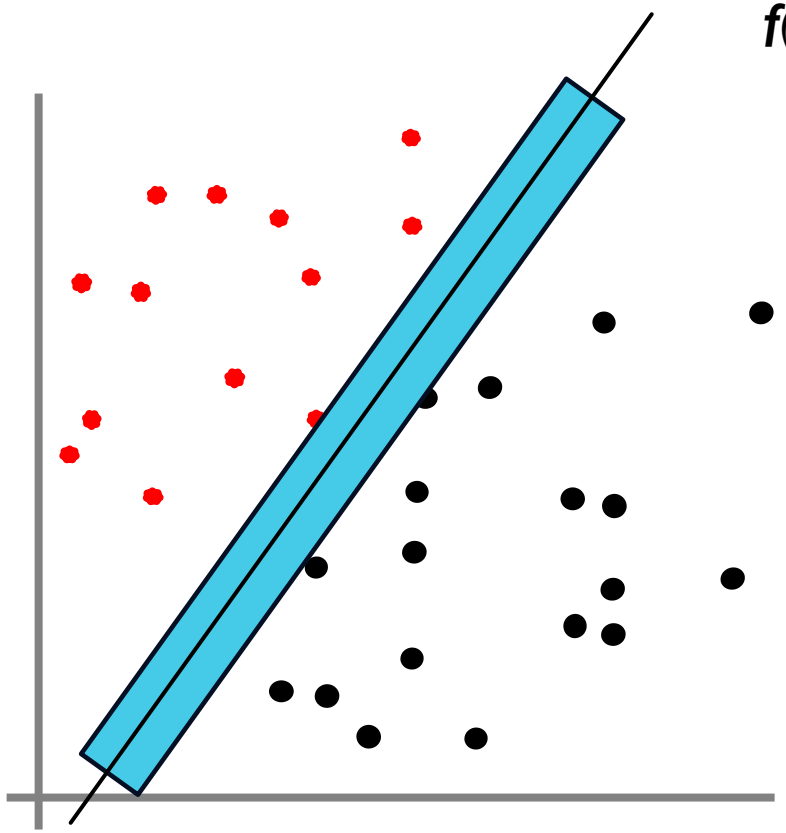
MAXIMAL MARGIN HYPERPLANE



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} - b)$

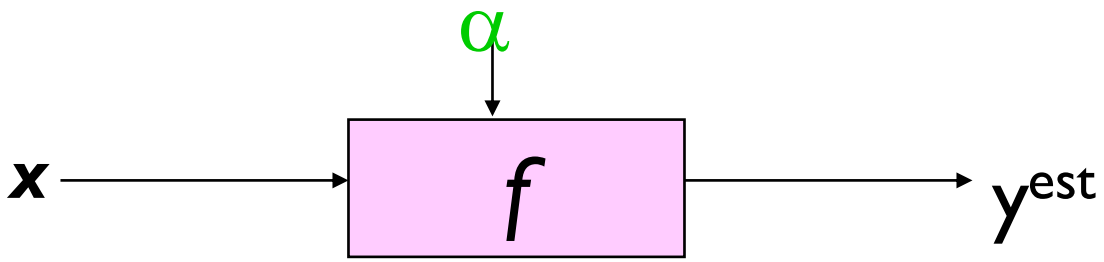
denotes +1

denotes -1



Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

LINEAR CLASSIFIERS

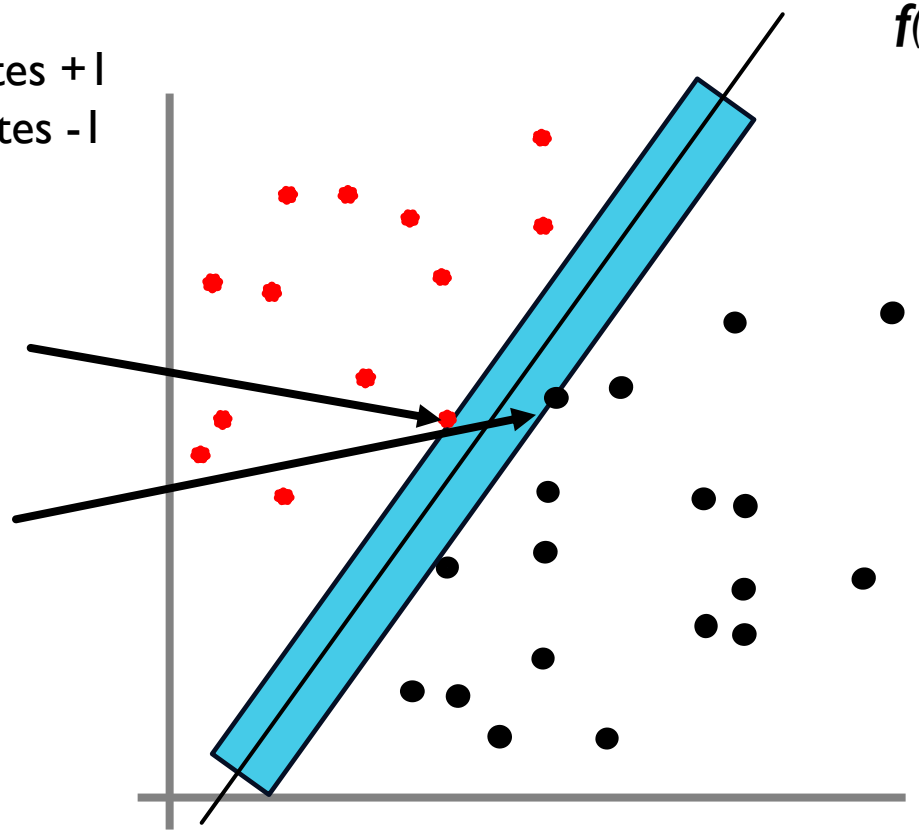


$f(x,w,b) = sign(w.x - b)$

denotes +1

denotes -1

Support Vectors are those datapoints that the margin pushes up against

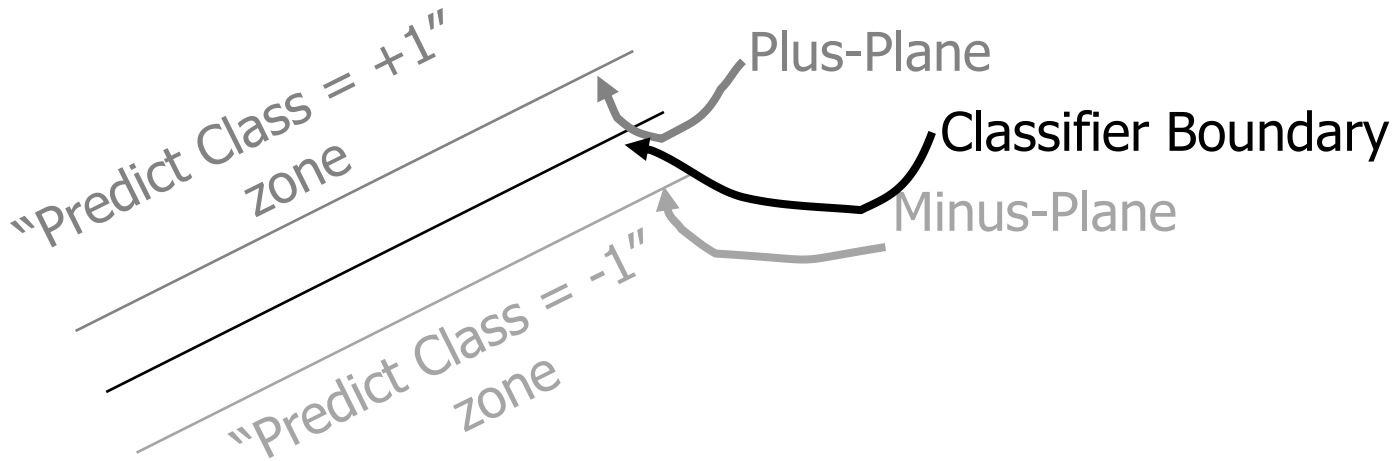


The maximum margin linear classifier is the linear classifier with the, maximum margin.

This is the simplest kind of SVM (Called an LSVM)

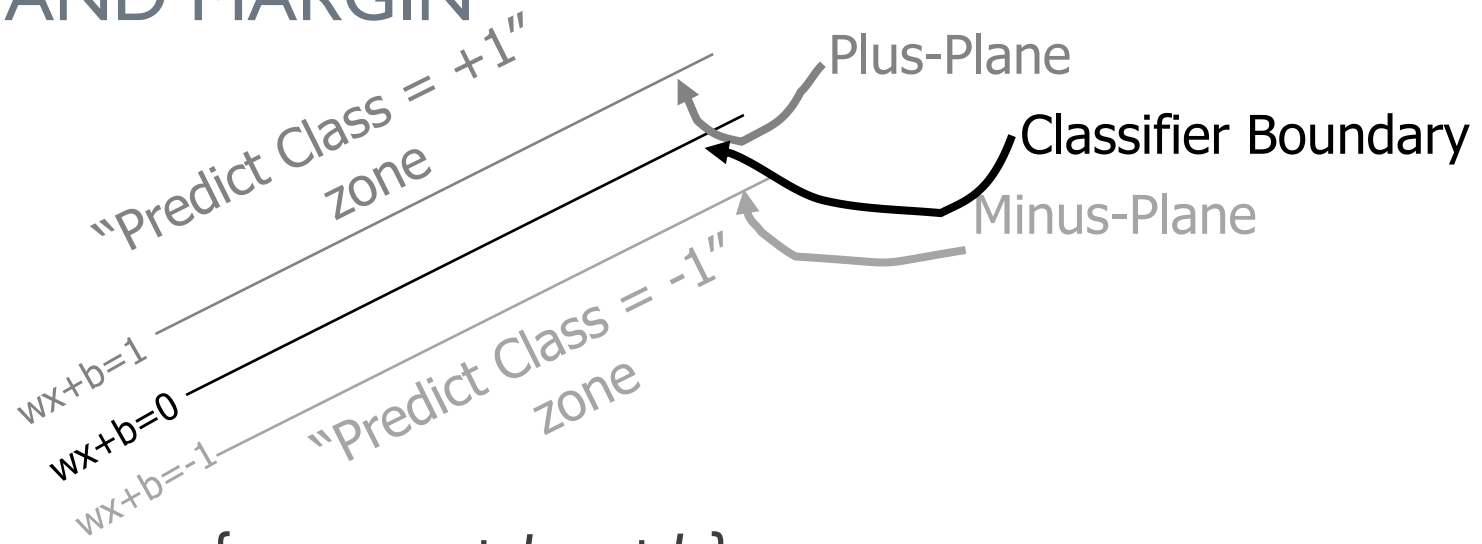
Linear SVM

SPECIFYING A LINE AND MARGIN



- How do we represent this mathematically?
- …in *m* input dimensions?

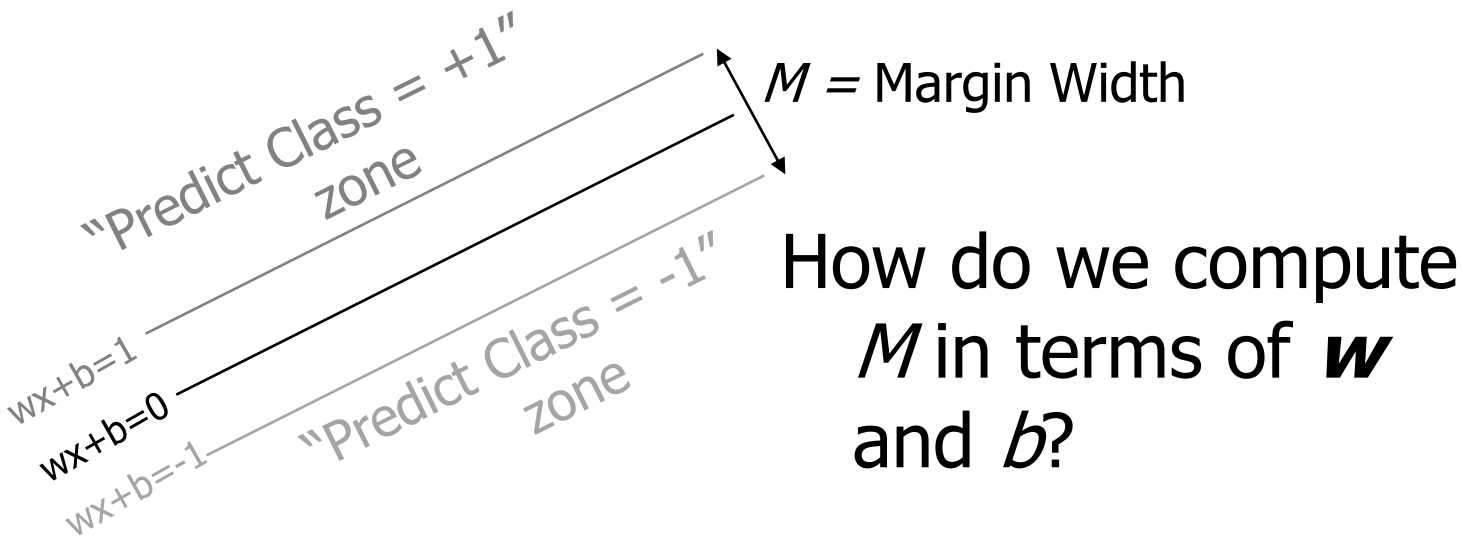
SPECIFYING A LINE AND MARGIN



- Plus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = +1 \}$
- Minus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = -1 \}$

Classify as..	+1	if	$\boldsymbol{w} \cdot \boldsymbol{x} + b >= 1$
	-1	if	$\boldsymbol{w} \cdot \boldsymbol{x} + b <= -1$
	Universe explodes	if	$-1 < \boldsymbol{w} \cdot \boldsymbol{x} + b < 1$

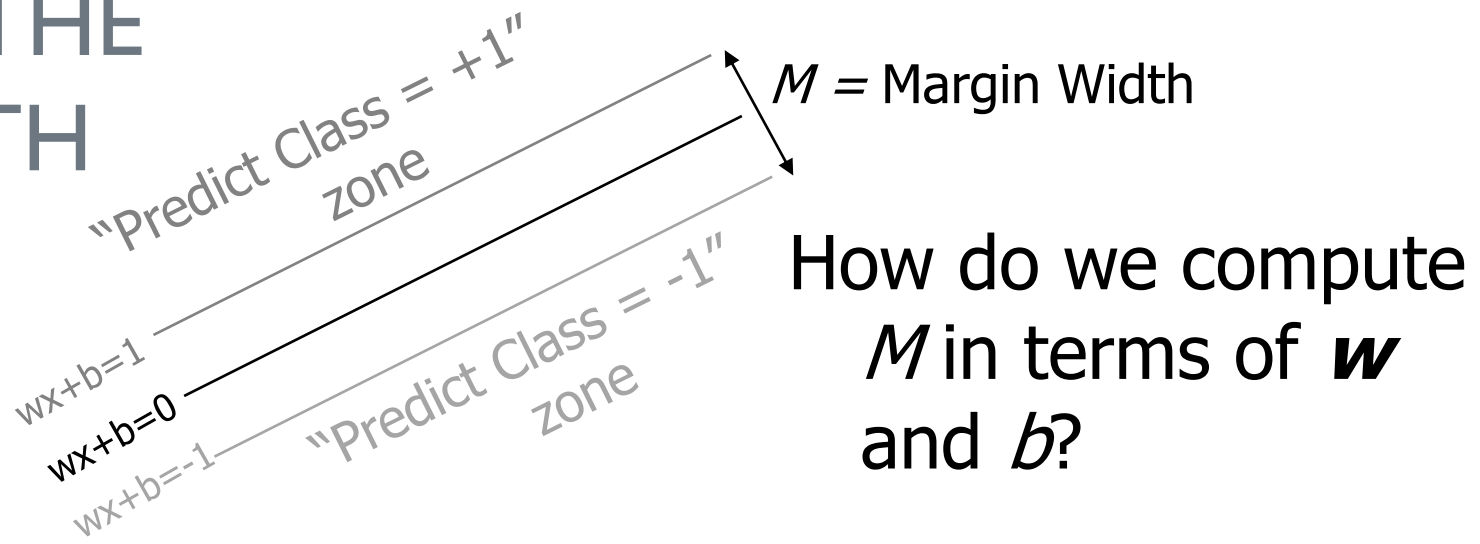
COMPUTING THE MARGIN WIDTH



How do we compute *M* in terms of ***w*** and *b*?

- Plus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = +1 \}$
- Minus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = -1 \}$

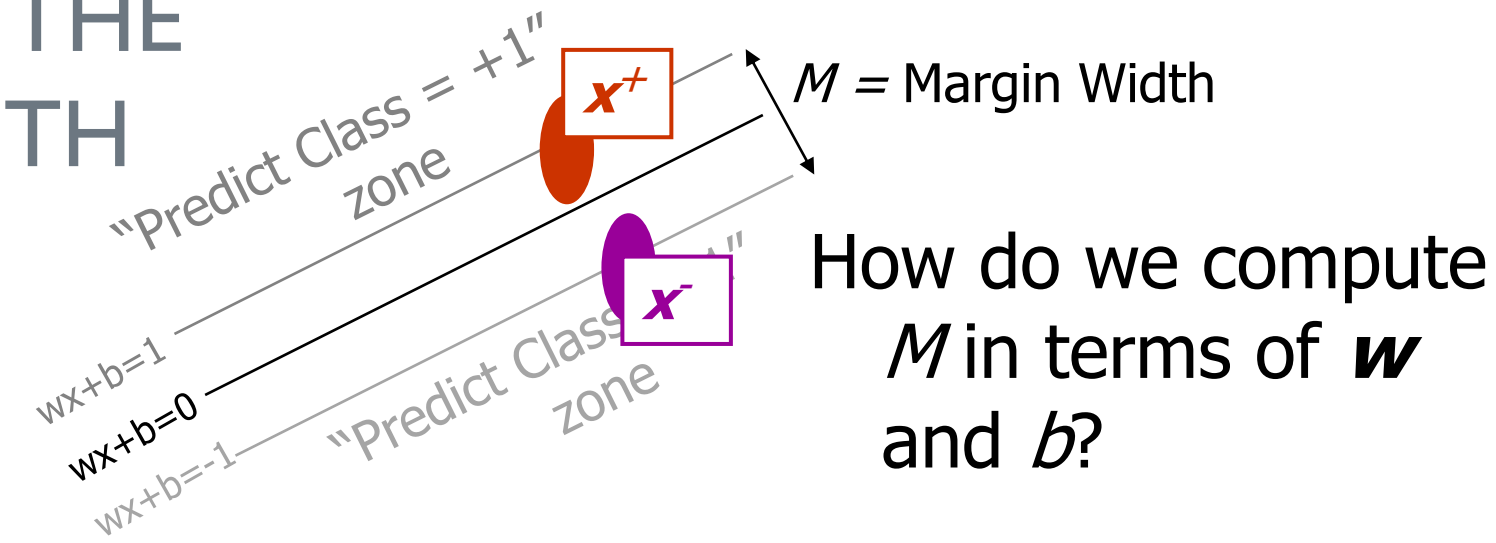
COMPUTING THE MARGIN WIDTH



How do we compute *M* in terms of ***w*** and *b*?

- Plus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = +1 \}$
- Minus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = -1 \}$

COMPUTING THE MARGIN WIDTH

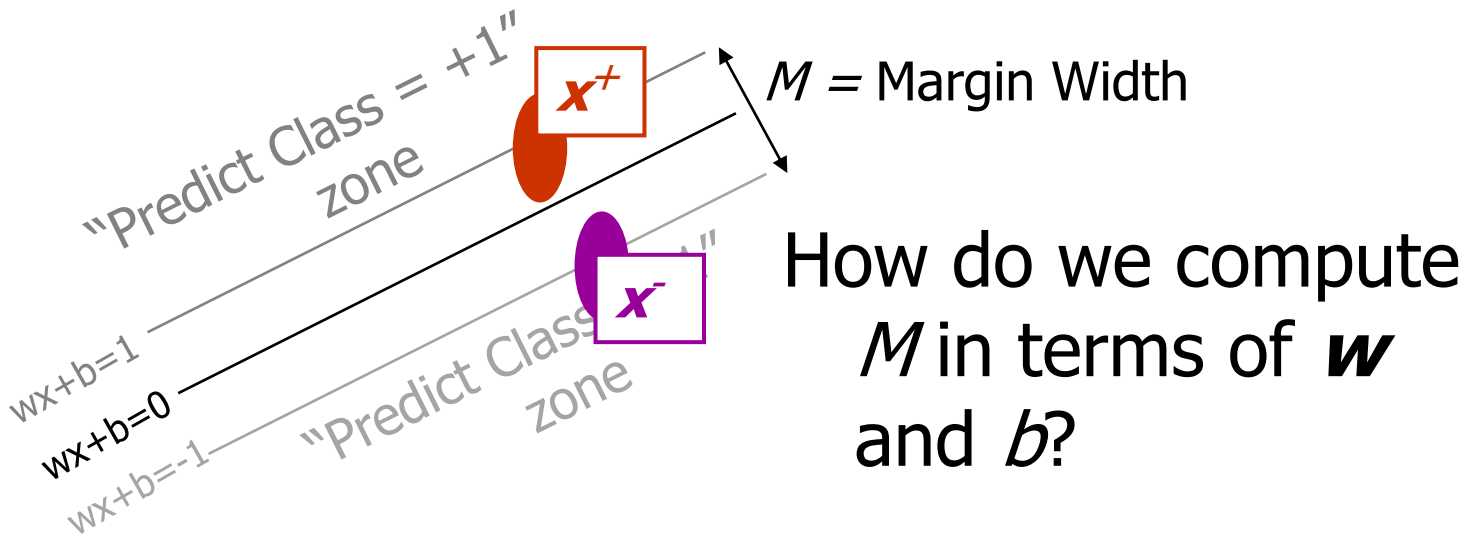


How do we compute *M* in terms of ***w*** and *b*?

- Plus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = +1 \}$
- Minus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = -1 \}$
- The vector **w** is perpendicular to the Plus Plane
- Let $\boldsymbol{x}^-$ be any point on the minus plane
- Let $\boldsymbol{x}^+$ be the closest plus-plane-point to $\boldsymbol{x}^-$.

Any location in R^m: not necessarily a datapoint

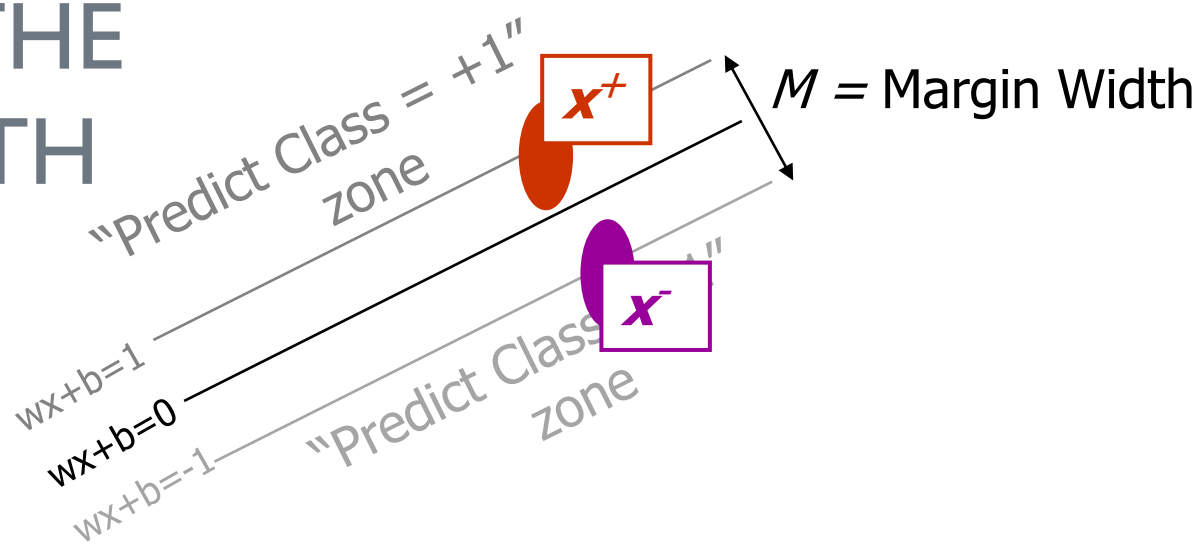
COMPUTING THE MARGIN WIDTH



How do we compute *M* in terms of ***w*** and *b*?

- Plus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = +1 \}$
- Minus-plane = $\{ \boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{x} + b = -1 \}$
- The vector **w** is perpendicular to the Plus Plane
- Let $\boldsymbol{x}^-$ be any point on the minus plane
- Let $\boldsymbol{x}^+$ be the closest plus-plane-point to $\boldsymbol{x}^-$.

COMPUTING THE MARGIN WIDTH

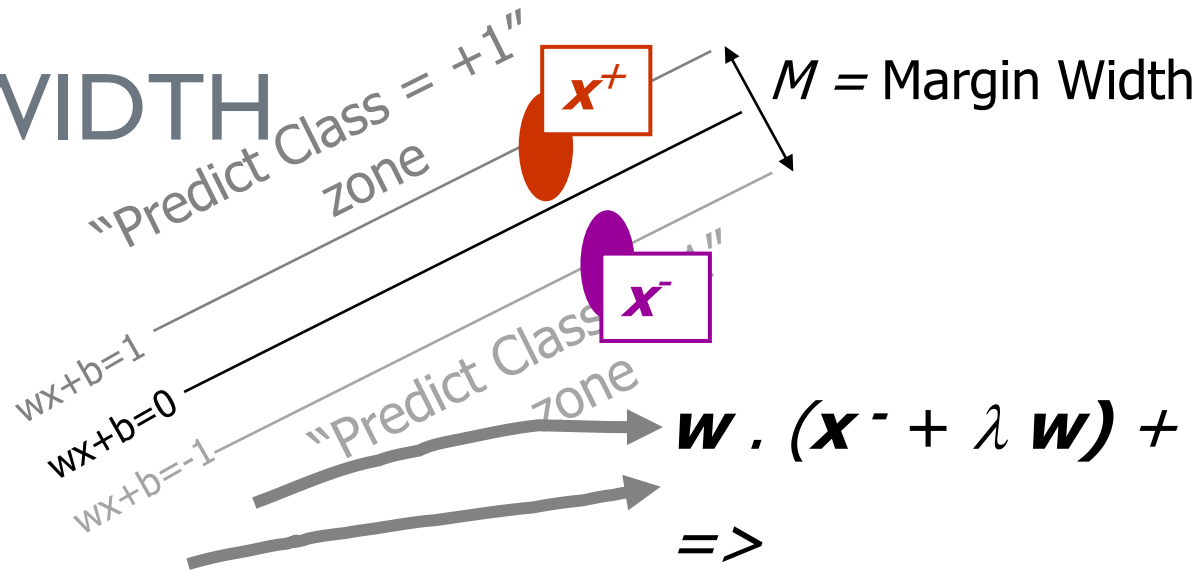


What we know:

- $\boldsymbol{w} \cdot \boldsymbol{x}^+ + b = +1$
- $\boldsymbol{w} \cdot \boldsymbol{x}^- + b = -1$
- $\boldsymbol{x}^+ = \boldsymbol{x}^- + \lambda \boldsymbol{w}$
- $|\boldsymbol{x}^+ - \boldsymbol{x}^-| = M$

It's now easy to get M in terms of $\boldsymbol{w}$ and b

COMPUTING THE MARGIN WIDTH



What we know:

- $w \cdot x^+ + b = +1$
- $w \cdot x^- + b = -1$
- $x^+ = x^- + \lambda w$
- $|x^+ - x^-| = M$

It's now easy to get M in terms of w and b

$$w \cdot (x^- + \lambda w) + b = 1$$

$$\Rightarrow$$

$$w \cdot x^- + b + \lambda w \cdot w = 1$$

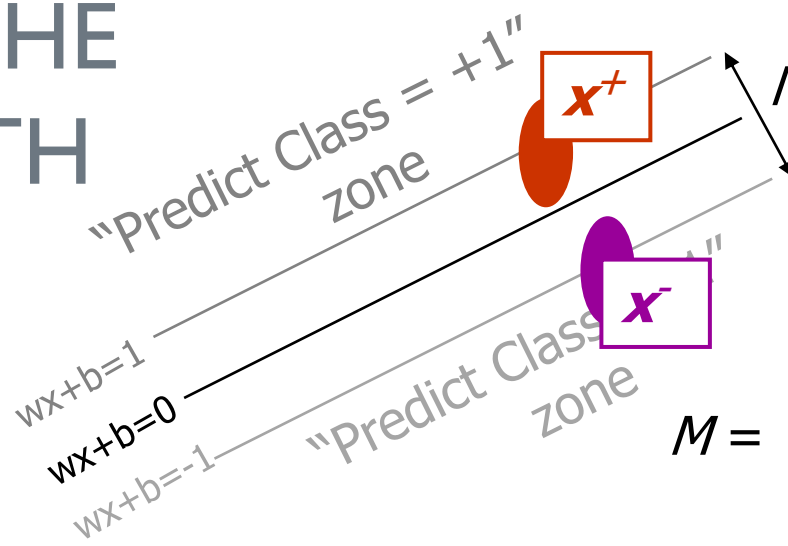
$$\Rightarrow$$

$$-1 + \lambda w \cdot w = 1$$

$$\Rightarrow$$

$$\lambda = \frac{2}{w \cdot w}$$

COMPUTING THE MARGIN WIDTH



M = Margin Width = $\frac{2}{\sqrt{\mathbf{w}.\mathbf{w}}}$

What we know:

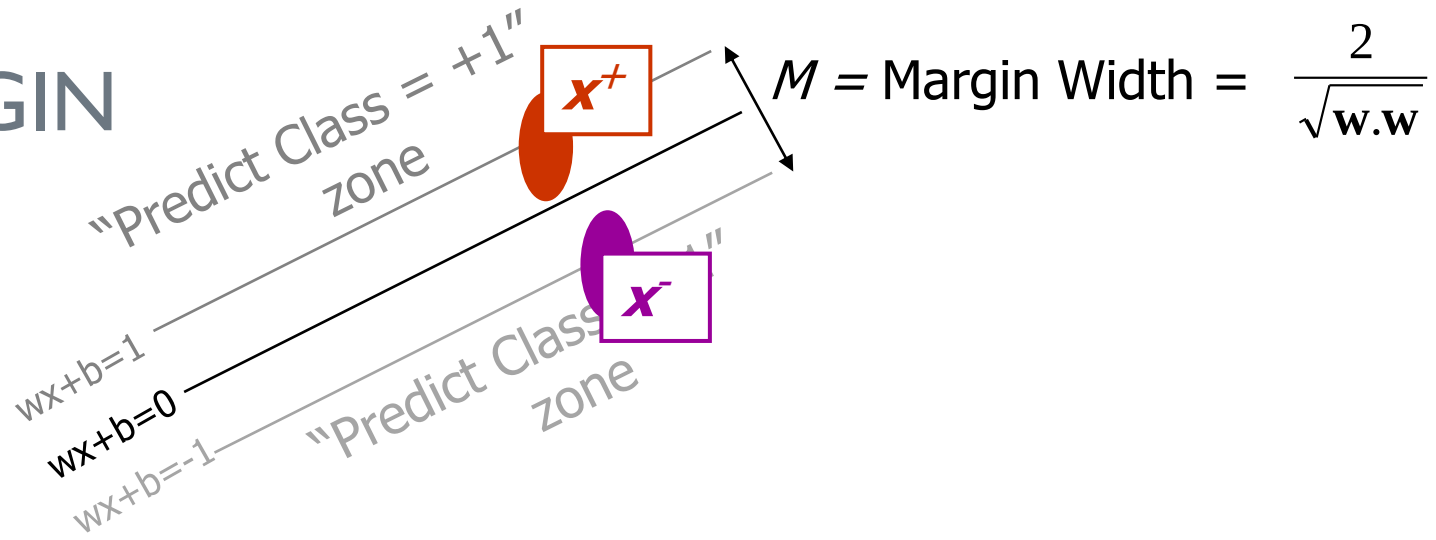
- $\boldsymbol{w} \cdot \boldsymbol{x}^+ + b = +1$
- $\boldsymbol{w} \cdot \boldsymbol{x}^- + b = -1$
- $\boldsymbol{x}^+ = \boldsymbol{x}^- + \lambda \boldsymbol{w}$
- $|\boldsymbol{x}^+ - \boldsymbol{x}^-| = M$
- $\lambda = \frac{2}{\mathbf{w}.\mathbf{w}}$

$$M = |\boldsymbol{x}^+ - \boldsymbol{x}^-| = |\lambda \boldsymbol{w}| =$$

$$= \lambda |\mathbf{w}| = \lambda \sqrt{\mathbf{w}.\mathbf{w}}$$

$$= \frac{2\sqrt{\mathbf{w}.\mathbf{w}}}{\mathbf{w}.\mathbf{w}} = \frac{2}{\sqrt{\mathbf{w}.\mathbf{w}}}$$

LEARNING THE MAXIMUM MARGIN CLASSIFIER



Given a guess of ***w*** and *b* we can

- Compute whether all data points in the correct half-planes
- Compute the width of the margin

So now we just need to write a program to search the space of **w**'s and *b*'s to find the widest margin that matches all the datapoints.

THE MATH

- Training instances
  - $x \in \Re^n$
  - $y \in \{-1, 1\}$
- Decision function
  - $f(x) = \text{sign}(\langle w,x \rangle + b)$
  - $w \in \Re^n$
  - $b \in \Re$
- Find w and b that
  - Perfectly classify training instances
    - Assuming linear separability
  - Maximize margin

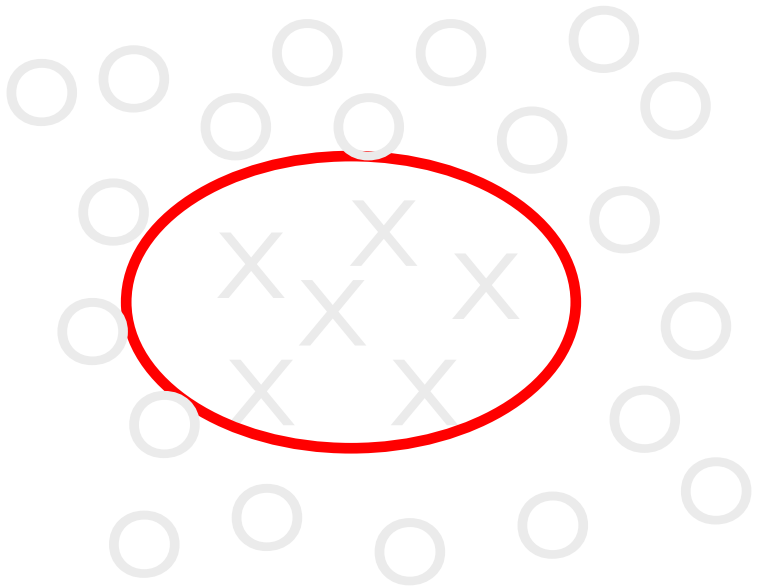
THE MATH

- For perfect classification, we want
  - y_i ($\langle w, x_i \rangle + b$) ≥ 0 for all i
  - Why?
- To maximize the margin, we want
  - w that minimizes $|w|^2$

STRENGTHS OF SVMS

- Good generalization in theory
- Good generalization in practice
- Work well with few training instances
- Efficient algorithms

WHAT IF SURFACE IS NON-LINEAR?



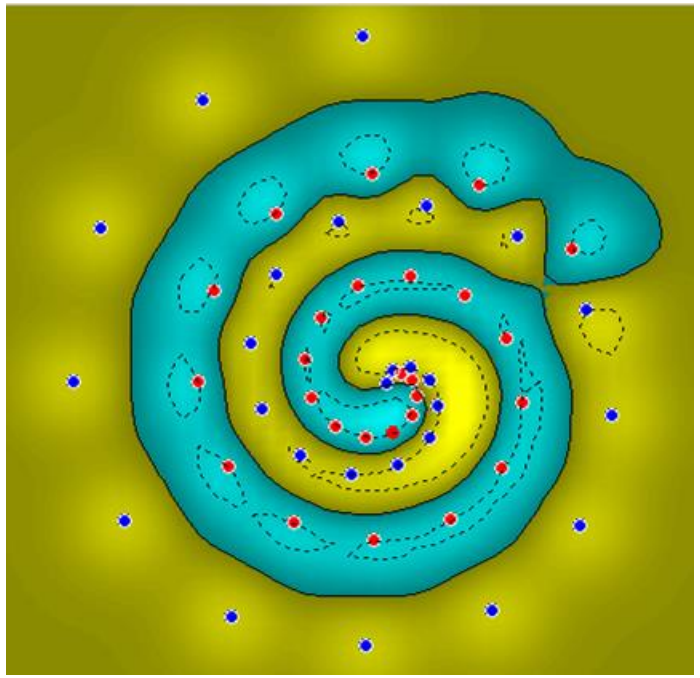
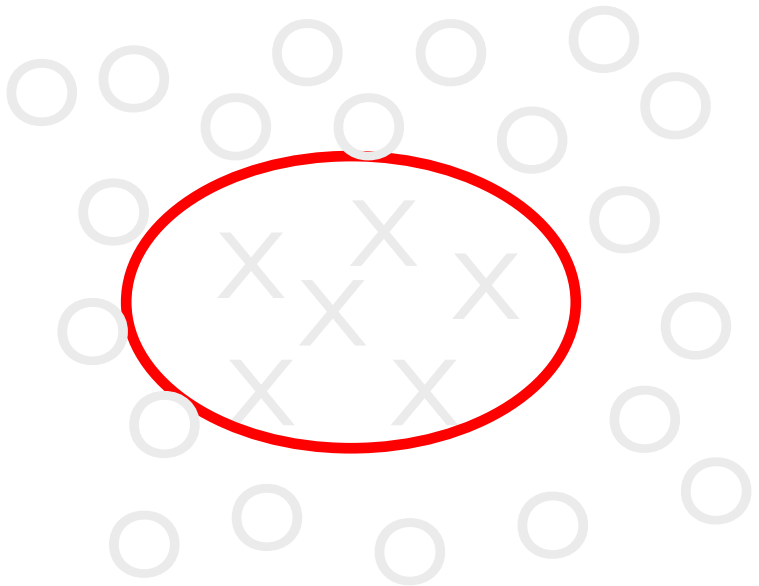


Image from http://www.atrandomresearch.com/iclass/

WHAT IF SURFACE IS NON-LINEAR?



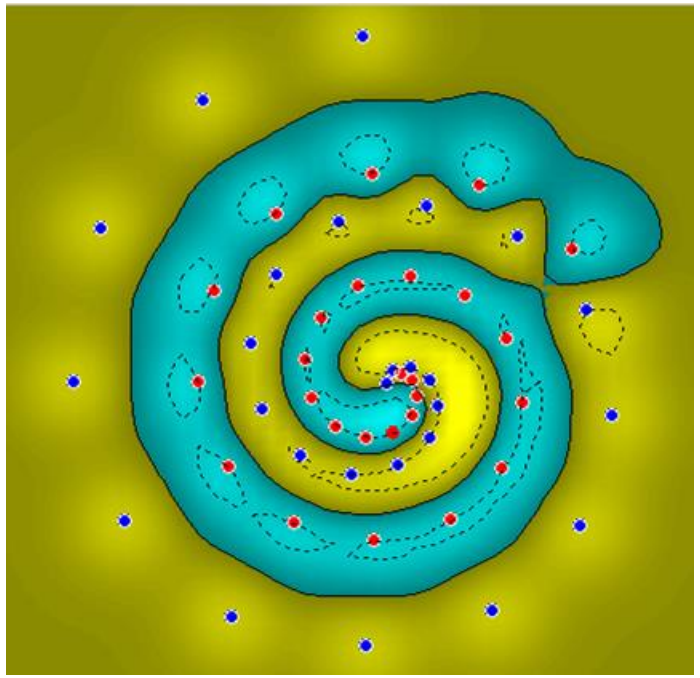
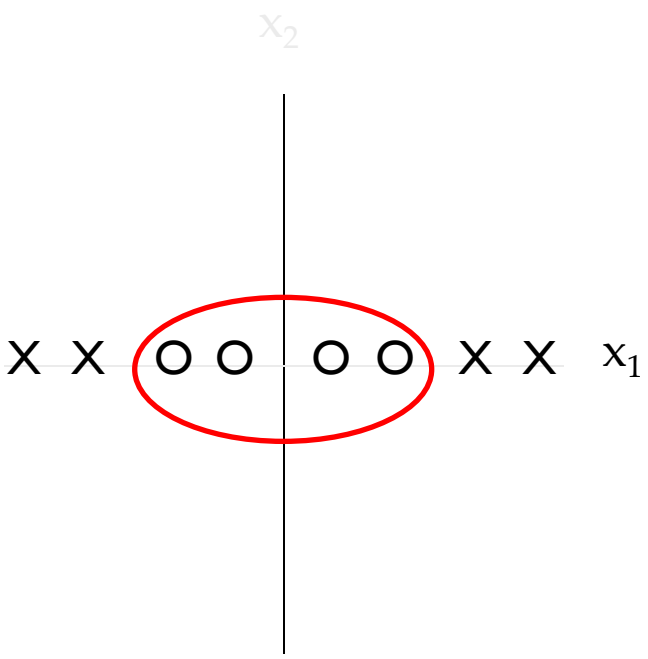
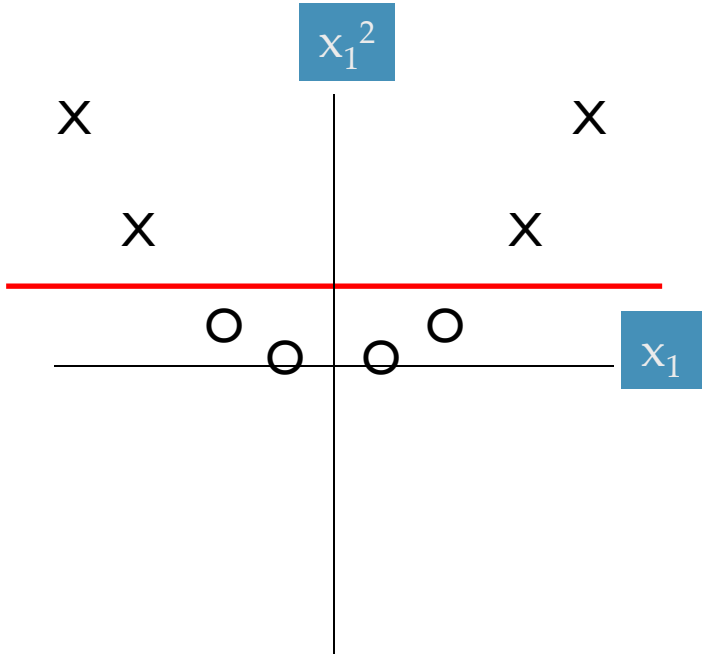


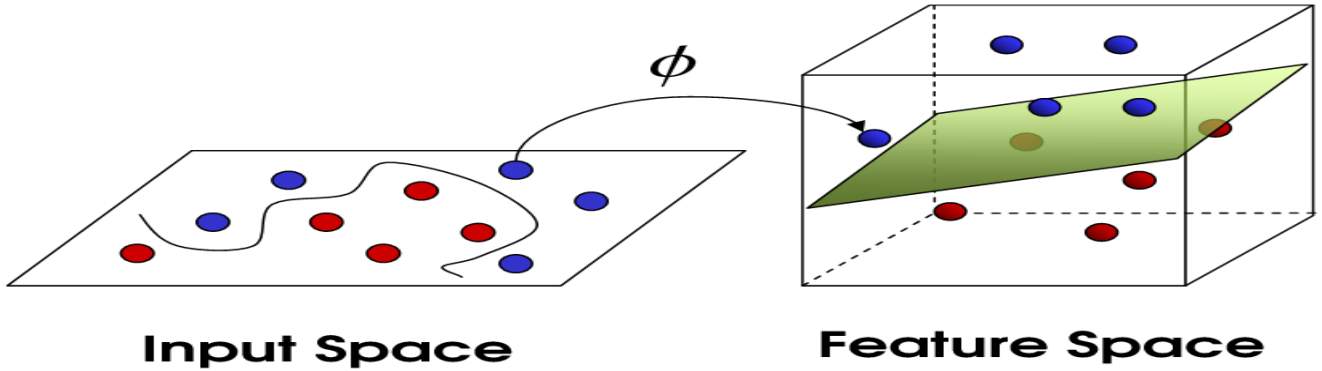
Image from http://www.atrandomresearch.com/iclass/

WHEN LINEAR SEPARATORS FAIL





MAPPING INTO A NEW FEATURE SPACE



$$\Phi : x \rightarrow X = \Phi(x)$$

$$\Phi(x_1,x_2) = (x_1,x_2,x_1^2,x_2^2,x_1x_2)$$

- Rather than run SVM on x_i, run it on $\Phi(x_i)$
- Find non-linear separator in input space
- What if $\Phi(x_i)$ is really big?
- Use kernels to compute it implicitly!

KERNELS

- Find kernel K such that
  - $K(x_1,x_2) = < \Phi(x_1), \Phi(x_2)>$
- Computing $K(x_1,x_2)$ should be efficient, much more so than computing $\Phi(x_1)$ and $\Phi(x_2)$
- Use $K(x_1,x_2)$ in SVM algorithm rather than $<x_1,x_2>$
- Remarkably, this is possible

THE POLYNOMIAL KERNEL

- $K(x_1,x_2) = < x_1, x_2 >^2$
  - $x_1 = (x_{11}, x_{12})$
  - $x_2 = (x_{21}, x_{22})$
- $< x_1, x_2 > = (x_{11}x_{21} + x_{12}x_{22})$
- $< x_1, x_2 >^2 = (x_{11}^2 x_{21}^2 + x_{12}^2 x_{22}^2 + 2x_{11} x_{12} x_{21} x_{22})$
- $\Phi(x_1) = (x_{11}^2, x_{12}^2, \sqrt{2}x_{11} x_{12})$
- $\Phi(x_2) = (x_{21}^2, x_{22}^2, \sqrt{2}x_{21} x_{22})$
- $K(x_1,x_2) = < \Phi(x_1), \Phi(x_2) >$

THE POLYNOMIAL KERNEL

- $\Phi(x)$ contains all monomials of degree d
- Useful in visual pattern recognition
- Number of monomials
  - 16x16 pixel image
  - 10^{10} monomials of degree 5
- Never explicitly compute $\Phi(x)$!
- Variation - $K(x_1,x_2) = (< x_1, x_2 > + 1)^2$

A FEW GOOD KERNELS

- Dot product kernel
  - $K(x_1,x_2) = < x_1,x_2 >$
- Polynomial kernel
  - $K(x_1,x_2) = < x_1,x_2 >^d$ (Monomials of degree d)
  - $K(x_1,x_2) = (< x_1,x_2 > + 1)^d$ (All monomials of degree 1,2,…,d)
- Gaussian kernel
  - $K(x_1,x_2) = \exp(-| x_1-x_2 |^2/2\sigma^2)$
  - Radial basis functions
- Sigmoid kernel
  - $K(x_1,x_2) = \tanh(< x_1,x_2 > + \vartheta)$
  - Neural networks
- Establishing "kernel-hood" from first principles is non-trivial

THE KERNEL TRICK

"Given an algorithm which is formulated in terms of a positive definite kernel K_1, one can construct an alternative algorithm by replacing K_1 with another positive definite kernel K_2"

- SVMs can use the kernel trick

EXOTIC KERNELS

- Strings
- Trees
- Graphs
- The hard part is establishing kernel-hood

CONCLUSION

- SVMs find optimal linear separator
- The kernel trick makes SVMs non-linear learning algorithms

THANK YOU